

DATA PROCESSING DEVICE FOR ERROR DETECTION OF UNDEFINED LENGTH INSTRUCTION

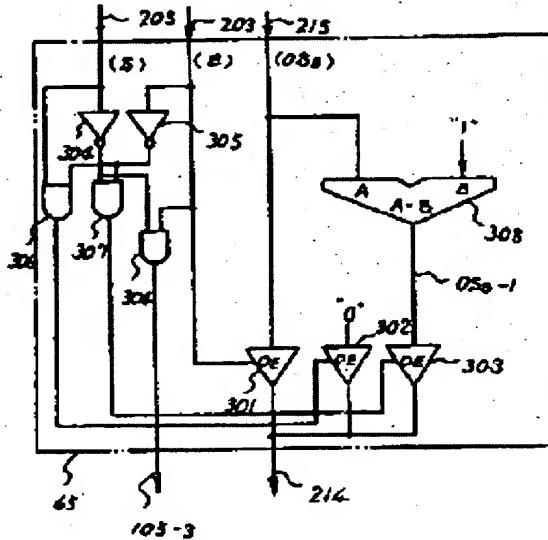
Patent number: JP58035643
Publication date: 1983-03-02
Inventor: FUKUNAGA YASUSHI; BANDOU TADAALKI; HIRAOKA YOSHINARI; HIRASAWA KOUTAROU; MATSUMOTO HIDEKAZU; IDE TOSHIYUKI; KATOU TAKESHI; NAKANISHI HIROAKI; KAWAKAMI TETSUYA
Applicant: HITACHI LTD;; HITACHI ENG CO LTD
Classification:
 - **international:** G06F9/30; G06F9/36
 - **european:** G06F9/30T; G06F9/30T2; G06F9/318; G06F11/08
Application number: JP19810132717 19810826
Priority number(s): JP19810132717 19810826

[Report a data error here](#)

Abstract of JP58035643

PURPOSE: To improve the capability of error detection for undefined length instructions, by detecting errors on a basis of contents of the end flag, which is added to an operand designator, and the number of operands.

CONSTITUTION: When an end flag E of an operand inputted to a decode processing end byte number detector 65 is (1), an output gate 301 is opened, and the number of bytes OSB of the operand designator on a bus 215 is outputted to a bus 214. In this case, unless a stop bit S of the operand designator is (1), it means that the number of operand designator is larger than the number of operands, and this is detected as an error. Then, an AND gate 304 is turned on to output a signal 105-3, and the error occurrence is transmitted to an address calculating unit of a CPU. Thus, the capability of error detection for undefined length instructions where the operand designator is used commonly for plural operands to be processed is improved.



Data supplied from the **esp@cenet** database - Worldwide

THIS PAGE BLANK (USPTO)

⑨ 日本国特許庁 (JP) ⑪ 特許出願公開
 ⑫ 公開特許公報 (A) 昭58-35643

⑩ Int. Cl.³
 G 06 F 9/30
 9/36
 11/08

識別記号 庁内整理番号
 6745-5B
 6745-5B
 7257-5B

⑬ 公開 昭和58年(1983)3月2日
 発明の数 1
 審査請求 未請求
 (全 12 頁)

⑭ 不定長命令のエラー検出を行うデータ処理装置
 ⑮ 特願 昭56-132717
 ⑯ 出願 昭56(1981)8月26日
 ⑰ 発明者 福永泰
 日立市幸町3丁目1番1号株式
 会社日立製作所日立研究所内
 ⑱ 発明者 坂東忠秋
 日立市幸町3丁目1番1号株式
 会社日立製作所日立研究所内
 ⑲ 発明者 平岡良成
 日立市幸町3丁目1番1号株式

会社日立製作所日立研究所内
 ⑳ 発明者 平沢宏太郎
 日立市幸町3丁目1番1号株式
 会社日立製作所日立研究所内
 ㉑ 発明者 松本秀和
 日立市幸町3丁目1番1号株式
 会社日立製作所日立研究所内
 ㉒ 出願人 株式会社日立製作所
 東京都千代田区丸の内1丁目5
 番1号
 ㉓ 代理人 弁理士 高橋明夫
 最終頁に続く

明細書

発明の名称 不定長命令のエラー検出を行うデータ処理装置

特許請求の範囲

1. オペランドのアドレスシングモードを指定するオペランド指定子が処理の種類およびオペランドの数を指定するオペコード部分から独立して命令を扱うデータ処理装置において、該命令は、少なくともオペランド指定子に対応して付加されただオペランド指定子の終了情報を含み、デコード手段は、少なくともオペコードデコード手段と、オペランド指定子デコード手段とオペランド指定子に付加された終了情報の検出を行う終了フラグ検出手段を有し、該オペコードデコード手段が、当該命令の最後のオペランドであることを示す信号を出力したとき、該終了情報検出手段が終了情報のセットを検出しない場合、当該命令をエラーとして検出するようにしたことを特徴とする不定長命令のエラー検出を行うデータ処理装置。

発明の詳細な説明

本発明は、オペランドのアドレスシングモードを指定するオペランド指定子が処理の種類およびオペランド数を指定するオペコード部分から独立して命令を扱うデータ処理装置に関するものである。

オペランド指定子の長さは、アドレスシングモードに対応して任意に変り、命令の長さが可変であることからこのようないきを可変長命令と呼ぶこともある。

「不定長命令」と「可変長命令」との間には特別の意味上の差異はなく、「不定長命令」なる用語を「可変長命令」なる用語に置き換える同一の意味を持つ。しかし、ここでは、便宜上、本願発明が扱う命令を不定長命令、従来の命令を可変長命令と称している。

可変長のオペランド指定子を持つ命令体系として、公知の代表的な2つの例を次に示す。

1つはバローズ社 (Burroughs Corporation) の計算機 B1700 を COBOL/RPG 向きのアーキテクチャとした時の命令フォーマットであり、こ

れは、「B1700 COBOL/RPG-S-Language, 1058823-015, Copyright 1973, Burroughs Corporation」に示されている。

今1つの例は、DEC社(Digital Equipment Corporation)の計算機VAX 11/780のアーキテクチャが有する可変長なるオペラント指定子を持つ命令体系であり、これは、「VAX11 Architecture Handbook, Copyright 1979」およびUSP #4236206に示されている。

ここに示した従来の2つの命令体系では、オペラントの形式、アドレスシングモードを指定する部分が、可変長のオペラント指定子で規定され、オペコードから独立であるといった特徴がある。

しかしながら、従来の可変長命令では、処理するオペラントの数と、処理するオペラントのアドレスシングモードを指定するオペラント指定子が1対1に対応づけられており、例えば

$A + B \rightarrow B$

$A + B \rightarrow C$

このような従来例にあつては、命令のエラー検出が困難であつた。

本発明の目的は、複数個の処理するオペラント数に対し、オペラント指定子を共用できる不定長命令のエラー検出能力を向上させたデータ処理装置を提供するにある。

本発明の特徴は、オペラントのアドレスシングモードを指定する各オペラント指定子にそれぞれオペラント指定子の終了情報(以下フラグと呼ぶ)を付加し、この終了フラグの内容と、オペコードが指示するオペラントの数によりエラー検出を行うよう正在していることである。

つまり、 $A + B \rightarrow B$ という処理では、2つのオペラントを処理し、2つ目のオペラントに対しては、これを2度使用するということを、オペコードで特別に規定する必要があるからである。

もし、 $A + B \rightarrow B$ という処理で、オペラント数を3とし、オペラント指定子を3つ用意すれば、 $A + B \rightarrow B$ と、 $A + B \rightarrow C$ の区別を意識する必要はないが、 $A + B \rightarrow B$ の処理では、全く同一のオ

特開昭58-35643(2)

という2つの処理(オペレーション)に対し、2つのオペコードを割当てる必要があつた。

具体的には、後述するが、オペコードデコード手段がオペラントの最終であることを示す信号を出力した時に、終了フラグのセットが検出されない場合当該命令をエラーとして検出するよう正在していることである。

以下、まず、本願発明の対象である不定長命令を扱うデータ処理装置を図面を参照して詳細に説明する。

第1図は本発明が適用されるデータ処理システムの基本的な概念図である。

メモリ装置1と、複数の中央処理装置2が共通バス3で接続されており、共通バス3を介して相互の間の情報交換が可能となつて正在。

メモリ装置1は、命令および命令が扱うオペラントを格納するメモリ部11と、この命令およびオペラントの読み出し、書き込みを制御するメモリ制御部12で構成され、メモリ部11とメモリ制御部12はメモリバス13で接続されている。

オペラント指定子を2つ設ける必要があり、これは、オペラント指定子そのものが、複数バイト(一般に長いものは7バイト)をとる時、メモリの実装効率を著しく低下し、好ましくない。

そもそも、 $A + B \rightarrow B$ と、 $A + B \rightarrow C$ の処理を区別することは、メモリの実装効率を高める為になされたものである。($A + B \rightarrow B$ ではオペラント指定子を2つで済むようにした)

このように、従来の方式では、同一のオペラントを複数回使用する処理に対しては、他の同機能の処理と区別する必要があり、オペコードで指定可能な処理数に制限があつた。

尚 $A + B \rightarrow B$ という例で説明したが、これは $A - B \rightarrow B$ という処理も同様であり、AとBの演算をし、その結果をBに格納するという例の全てにいえることである。一般にこれを $A \oplus B \rightarrow B$ と表現している。

メモリ装置1の動作については、特開昭55-160758号明細書に詳細に記載しているが、本発明の要部とは直接関係ないので、この部分の詳

細説明は省略する。

中央処理装置 2 は、共通バス 3 に複数台接続可能で（図示では 2 台）、それぞれ、メモリ装置 1 より命令及びオペラントをアクセスして順次命令の処理をしていく。

ここでは高速化のため、一度読み出された命令、オペラントをそれぞれコピーしている命令キヤツシユ 2 1（高速バッファメモリ）およびオペラントキヤツシユ 2 2（高速バッファメモリ）を有し、また命令のフェッチ、デコード及びオペラントアドレス演算を行う I ユニット 2 3 と、オペラントのフェッチおよび命令の実行を行う E ユニットを有し、それぞれがパイプライン処理を行う例を示している。

このような、命令キヤツシユ、オペラントキヤツシユの使い方、或いは、I ユニットと E ユニットがパイプライン処理すること自体は公知である。

さて、第 2 図 (A) は、中央処理装置 2 が扱う不定長命令のフォーマットを示している。

1 つの命令は、1 ないし数バイトで構成される

ラントが繰返し、使用される。

同一のオペラントを繰返し使用するには、いろいろな実現方法があるが、最も望ましい方法は、最後のオペラント指定子を繰返し使用することであろう。この点については後で詳細に述べる。

オペコード O P で指定するオペラント数と、オペラント指定子の数が不一致の例を次に説明する。

例えば、オペコード O P が加算処理の場合、そのファンクションは、

$A + B \rightarrow C$

で、3 つのオペラントを必要とするが、オペラント指定子が 1 つの場合は、

$A + A \rightarrow A$

2 つの場合は、

$A + B \rightarrow B$

という処理が、同一のオペコードで可能となる。オペラント指定子の具体的な例を第 2 図 (B) に示している。

ここでは、M 1 ～ M 4 の例を示し、そのフォーマットと、これに対応するオペラントを 1 対 1

特開昭58- 35643 (3)

オペレーションコード（これは通称、オペコードと呼ばれている）O P と、終了フラグ S を伴つた 1 ないし複数バイトのオペラント指定子 O S 1, O S 2, …, O S n から構成されている。

オペコード O P では、その命令の処理内容（処理の種類）、処理に必要なオペラントの数およびオペラントの属性（データ長、リード/ライトの区別、データタイプ：固定小数点/浮動小数点…等）が示される。

オペコード O P の後には、当該オペコード O P で示されるオペラントの数以下のオペラント指定子 (O S 1, O S 2, …) が示されていて、1 つの命令（正式には命令語：インストラクション・ワード）が構成される。

オペラント指定子は、該当命令で使用されるオペラントの順番に並んでいて、最後のオペラント指定子のみ、終了フラグ S が「1」にセットされている。もし、オペラント指定子の数が、オペコード O P で指定されるオペラント数より少ない場合は、最後のオペラント指定子に対応するオペ

対応で示している。

第 2 図 (B) において、オペラントの () は、() 内の値をアドレスとしたメモリの内容であることを示している。

また、フォーマット中、DISP は、変位を、IM はイミーディエット（データ直接）を示し、添字はその大きさをビット数で示している。

更に、R. はインデックスレジスタ、R. はジエネラルレジスタを示し、L はオペラントの大きさをバイト単位で示したものである。

第 2 図 (B) において、そのフォーマットとオペラントの関係は、或程度理解できると思われるが、以下、簡単に説明する。

M 1 は、レジスタ直接のアドレス指定で、R. で示されるジエネラルレジスタそのものが、直接オペラントとなるものである。

M 2 以下は、全て、メモリをオペラントとするもので、そのアドレス計算がオペラントの欄に示された形で行なわれる。

M 2 は、間接アドレス指定で、R. で示される

ジエネラルレジスタの内容がオペランドのアドレスとなるものである。

図3, 5, 7では、R_xで示されるジエネラルレジスタの内容に、DISPで示される値が加算されて、これがオペランドのアドレスになつている。

図4, 6, 8では、図3, 5, 7で求められたアドレスのメモリの内容が、オペランドのアドレスとなるものである。

図9～11は、イミーディエットデータで、IM₈, IM₁₀, IM₃₂の値そのものがオペランドとなつている。

図12～17は、ジエネラルレジスタR_xの代りにプログラムカウンタPCが使用されることが、図3～8とは異なるだけである。PCは、デコードするオペランド指定子の次のアドレスを保持している。

図18～24は、図3～8にさらにインデックスレジスタR_xの値が加算されることが異なり、またインデックスレジスタR_xの値は、オペランドのデータ長L分乗算された値が加算される。

27およびアドレス計算ユニット(AU)28部がこれに対応し、Eユニット24にはオペランドフェッチユニット(OFU)29、実行ユニット(EU)30が対応している。第1図では、Iユニット23とEユニット24がバイブライン処理を行う旨述べたが、それぞれのユニットは第3図に示すように更に命令フェッチユニットIFU25、デコードユニットDU27、アドレス計算ユニットAU28、オペランドフェッチユニットOFU29、実行ユニット30に分割され、それぞれがバイブライン処理をする例を示している。

しかしながら、本願発明の要旨は、このようないい處を除いて、バイブライン処理そのものとは直接関係ないので、バイブライン処理については詳細な説明は省略している。尚、バイブライン処理そのものは周知であるが、U.S.P.4,025,771号には、バイブライン高速信号プロセッサが示されている。

ところで、第3図において、命令フェッチユニット25は、命令を先行してフェッチするためのプログラムカウンタ50を有し、命令キャッシュ

これは、データ長にかかわらず、インデックスレジスタR_xの値を、先頭からの変位としてセットできるようにするため必要となる処理である。

つまり、L(データ長を示す)を乗算することにより、インデックスレジスタR_xは、データ長にかかわらず、先頭から何番目のデータであるかを示す値を入れておけばよいことになる。

例えば、インデックスレジスタR_xに「10」が入つていると、これは先頭から10番目のデータで、そのアドレスは、バイトの場合は10を加算(L=1)、ワードの場合は20を加算(L=2)、ロングワード(Long Word)では40を自動的に加算し(L=4)、ユーザはデータ長にかかわらず、インデックスレジスタR_xの値をセットできる。

第3図は、第1図の中央処理装置2のより具体的なブロック構成図である。

第1図におけるIユニット23は、第3図において、命令フェッチユニット(IFU)25、アライナ(ALIG)26、デコードユニット(DU)

21より次に実行されるであろう命令を先行して読み出す処理を行なつている。

アドレスライン100ICより、読み出したいアドレスが命令キャッシュ21に送られ対応する命令4バイト分がデータライン101により命令フェッチユニットIFU25に送出される。

命令キャッシュ21IC、対応する命令がつかつた場合は、共通バス3を介して、メモリ1から該当の命令を読み出し、この命令は命令キャッシュ21ICにストアされる。キャッシュの動作は周知であり、例えば、「A Guide to the IBM System/370 Model 168」に示されている。

命令4バイト分が命令フェッチユニット25に送出されると、プログラムカウンタ50はプラス4(+4.)され、次の命令の送出要求を命令キャッシュ21に出力する。

この動作は、命令フェッチユニットIFU25内にあるバッファ(図示せず)が満杯になるまで繰り返される。

命令フェッチユニットIFU25からは、バス

103を介して、あらかじめ読み出しておいた命令がアライナ(ALIG)26に送出される。

アライナ26は、デコードユニットDU27からの信号線102に指示されたバイト数だけソフト処理を行い、バス104に該当の命令を送出する。

信号線102の値を適当に操作することで、バス104には第5図に示すように、命令の第1オペラント指定子処理時には、左端にオペコードOPが、続いて第1番目のオペラント指定子が並ぶように、2番目以降のオペラント指定子処理時には、1バイトのダミーを置いてオペラント指定子が配置して出力される。上記制後は、後で詳しく説明する。

デコードユニット(DU)27は、アライナ26(ALIG)より送出されたオペコード及びオペラント指定子をデコードして、アドレス計算ユニットAU28へ下記の情報を送る。

(1) バス105を介してアドレスシングモードを送る。

(4) バス108を介してインデックスレジスタR₁のアドレス、

および

(5) バス116を介してアドレス演算に使用するプログラムカウンタの値を送る。

アドレス計算ユニットAU28は、バス105によつて示されたアドレスシングモードに従い、上記(a), (e)以外の時は、オペラントのアドレス計算を行い、バス109に計算後のアドレスを送出する。

一方、(a)の場合は、バス107の内容を、そのままバス113に送出し、(e)の場合はバス106の内容をバス109に送出する。

オペラントフェッチャニットOFU29は、上記(a), (e)以外の時は、送られたアドレスが示されているバス109の内容を、バス110に送出し、オペラントがリード時には、オペラントキャッシュ22にリード処理を要求する。

リードオペラントが、オペラントキャッシュ22からバス111IC送出されると、オペラント

特開昭58-35643(5)

アドレスシングモードには、先に説明の如く、次の(a)～(h)があり、このうち1つが指定される。

(a) : レジスタ直接 …… A61

(b) : R₁ …… A62

(c) : R₁ + DISPタイプ …… A63, 5, 7

(d) : R₁ + DISPインダイレクトタイプ …… A64, 6, 8

(e) : イミーディエクト …… A69, 10, 11

(f) : PC+DISPタイプ …… A12, 14, 16

(g) : PC+DISPインダイレクトタイプ …… A13, 15, 17

(h) : (b)～(d)でインデックス付タイプ ……

A618～24

尚A61～A624は第2図(B)に示すオペラント指定子フォーマットのA61～24ICに対応している。

(2) バス106を介して、DISPまたはイミーディエクトデータを32ビットで送る。

(3) バス107を介してジエネラルレジスタR₁のアドレスを送る。

フェッチャニットOFU29は、バス112を介して実行ユニットEU30に、読み出されたオペラントを出し、またオペラントがそろつた旨を連絡する。

オペラントがライト時は、実行ユニットEU30からの書き込みデータが、バス111に送出されるまで、オペラントフェッチャニットOFU29は、アドレスをバス110に送出し続ける。

一方、上記(a)に対しては、オペラントフェッチャニットOFU29は、アドレス計算ユニットAU28より送出されたレジスタアドレス113ICにより、自身が有するジエネラルレジスタ(図示せず)のアクセスを行う。(a)以外と異なるのは、メモリアクセスするか、レジスタをアクセスするかの差異のみである。

また、(e)に対しては、バス109の内容をそのままバス111に送出し、実行ユニットEU30ICオペラントがそろつた旨を連絡する。

また、実行ユニットEU30は、デコードユニットDU27からオペコードバス114を介して

送出されたマイクロプログラムの先頭アドレスを受信し、リード時は、バス112のオペランドを用いて、ライト時は、オペランド(データ)をバス111に出力して順次、命令の処理を行う。

また、命令が分歧命令の場合は、バス115を用いて、新たなプログラムカウンタ値を命令ユニットIFU25のプログラムカウンタ50や、後述するデコードユニットDU27内のDPレジスタ69にセットすると同時に、パイプライン処理で先行的に処理されていたオペランドの各ユニットにおける処理結果をキャンセルさせる。

以上が、1つのオペランド指定子に対する処理の概略で、各ユニット(25~30)は、パイプライン処理で、順次オペランド指定子の処理を並列に処理していく。

次に、本発明の要旨に関係するデコードユニット27について、具体例を示し詳細に説明する。

第4図は、第3図に示すデコードユニットDU27の具体的な実施例を示すプロック図である。

トはオペコードレジスタ64にセットされる。

オペコードレジスタ64の出力は、該当命令の実行ユニットEU30のマイクロプログラムの先頭アドレスを求めるオペコードデコードユニット61と、該当命令のオペランドに対する情報を有するオペランド情報ROM63に送られる。

ROM61の出力結果201は先頭アドレスレジスタ62にセットされ、オペコードバス114を介して、第1オペランドがオペランドフェッチユニットOFU29から実行ユニットEU30に渡されるのに同期して、EU30に送出される。

ROM63は、例えば第6図に示す構成で、その中には、第6図に示すような情報があらかじめ入力されており、オペコードと第何番目のオペランドの処理であるかの情報をアドレスとして読み出される。

すなわち、オペコードレジスタ64に、第1バイト目がセットされた時には、セレクタSEL81で、バス200側が選択されるため、オペコードをアドレスとして、その第1オペランドに関する情

DPレジスタ69は、デコードユニットDU27がデコードする命令の先頭を示しており、第1番目のオペランド指定子デコード時は、オペコードのアドレスを、第2番目以下のデコード時は、該当オペランド指定子の先頭-1のアドレスを示している。

上記アドレスは、バス102を介して第3図に示すアライナALIG26、命令フェッチユニットIFU25に送出されているため、バス104には、第5図に示すように、1バイト目には、第1オペランド指定子の読み出しの場合は、(A)に示すように、オペコードOP、第2オペランド指定子以下の読み出しの場合は、(B)に示すように、ダミーのデータ、第2バイト目には、終了フラグSを含んだオペランド指定子の先頭バイトが、第3バイト目から第7バイト目には、オペランド指定子のその他の情報が出力される。

バス204は、第何番目のオペランドの処理をしているかを示す情報で、本情報が全オペランド処理終了を示している時、バス104の第1バ

トが読み出される。

読み出された情報としては、

- (1) オペランドの属性、すなわち、リードオペランドであるか、ライトオペランドであるかの情報R/Wや、オペランドのデータ長L(バイト、ワード、ロングワード)を示す情報、
- (2) オペランドの最終であることを示すフラグ、および
- (3) 同一命令の次オペランドの情報が入っているアドレス、

がある。

(1)はバス105-1に出力され、アドレス計算ユニットAU28に出力され、また(2)は、バス203に出力され、デコード処理終了バイト検出器65に送出される。

また、(2)、(3)の情報は、レジスタ83にラッチされた後、バス204に出力され、次のオペランドを読み出すアドレスとして使用される。

(2)の情報のラッチ情報が、セレクタ81の選択

端子 S に入力されるため、(2)の情報が “1” の場合は、オペコードレジスタ 64 の内容 (200) が使用され、“0” の場合は(3)の情報が使用される。

一方、バス 104 の中で、終了フラグ S を示す信号線 205 は、デコード処理終了バイト数検出器 65 に送出される。

また、オペランド指定子の先頭 7 ビットは、バス 206 により、オペランド指定子デコーダ 66 に送られる。7 ビットの情報によりオペランド指定子のデコードを行うが、その例を第 7 図(A)により説明する。

たとえば、第 2 図 (B) の A3 に示す (R₁ + DISP₁) のオペランド指定子が送られると、第 7 図 (A) のように上位 7 ビットの中の更に 3 ビットが 010 であることを検出して、下記情報が output できる。

- (1) 2 バイト長のオペランド指定子であること、
- (2) バス 208 の内容をバス 106 へ出力する場合、DISP の桁合せを行うため 3 バイトの

- (5) R₁ の情報は、オペランド指定子の 2 バイト目の下位 4 ビットに存在すること、

の 5 つである。

以上 2 つの例につき示したがこれらをまとめてみると次のようになる。

オペランド指定子デコーダ 66 は、送られてきたオペランド指定子をデコードし、次にあげる情報をそれぞれ出力する。

- (1) バス 215 へは、オペランド指定子の長さをバイト単位で出力する。例えば、第 2 図(B) の A3 のオペランド指定子で、(R₁ + DISP₁) のオペランド指定子が送られてきた時は、“2”を出力する。
- (2) バス 211 へは、ディスプレースメント (DISP) / イミーデイエット (IM) データ用アライナ 67 に対するシフトバイト数を出力する。

例えば、(R₁ + DISP₁) のオペランド指定子の場合は、第 7 図 (A) に示した如く 3 バイト右シフトを、(R₁ + DISP₂) の場合

右シフトが必要であること、

- (3) DISP 値として 4 バイト化を図るため、上位 3 バイトについては DISP₁ の最上位ビットを符号拡張して出力すること、
- (4) R₁ + DISP でもつてオペランドのアドレスが計算できること、
- (5) R₁ の情報は 1 バイト目の下位 4 ビットに存在すること、

の 5 つである。同様に、第 2 図 (B) の A7 の (R₁ + DISP₂) が送られてくると、第 7 図 (B) に示すように上位 7 ビットが 1110110 であることを検出して、下記情報が output できる。

- (1) 6 バイト長のオペランド指定子であること、
- (2) バス 208 の内容をバス 106 へ出力する場合、DISP の桁合せを行うため 1 バイト左シフトが必要であること、
- (3) DISP は 32 ビット全てが指定されているためそのまま出力しなければならないこと、
- (4) R₁ + DISP でもつてオペランドのアドレスが計算できること、

は、第 7 図 (B) に示した如く 1 バイト左シフトを指示する。

- (3) バス 212 へは、アライナ 67 に対するマスクバイトの指示データを出力する。

これは、アライナ 67 に対し、バス 106 に output する 4 バイトのデータ中、上位 2 バイト、又は 3 バイトのマスクを指示することで、1 バイトや 2 バイトの DISP, IM 情報の符号拡張による 4 バイト化を図るためのものである。

例えば、DISP₁ の時には、第 7 図 (A) に示す如く 3 バイトシフト、DISP₂ の時は第 7 図 (B) の如く前に “— R₁” という余分な 1 バイトがあるため、1 バイト左シフトする。

これは、DISP₁ の時、上 3 バイトには DISP₁ の符号ビットを拡張して入れておかないと、32 ビットの正常なアドレス計算ができないからである。(バス 212 はその指定のためのもの)

(4) バス 105-2 へはアドレスシングモードを出力し、これによつてアドレス計算ユニット A U 28 の動作モードを指示する。

アドレスシングモードについては、第3図のアドレス計算ユニット A U 28 の説明に関連して、(a)～(h)の8つのモードがあることを既に説明した。

(5) バス 216 へはジエネラルレジスタ R。の存在する位置が、1 バイト目か 2 バイト目かを示す情報を出力する。

$(R_n + DISP_1)$ 時は 1 バイト目、 $(R_n + DISP_2)$ 時は 2 バイト目が指示される。一方、バス 108 へは、オペラント指定子の中のインデックスレジスタ R_n の部分を出力する。また、セレクタ 68 は、信号 216 で指定されたジエネラルレジスタ R_n の存在する位置 (1 バイト目か 2 バイト目かの信号) によつて、 R_n に対応する部分 (バス 207 の内容或いはバス 210 の内容) をバス 107 へ出力する。

アライナ 57 は、前述したように、オペラント

指定子の 2 バイト目から 7 バイト目までがバス 208 によつて与えられるため、信号線 211 で与えられたシフト数だけシフト処理を行い、且つ信号線 212 で与えられたマスク部に対しても、符号の拡張を行い、バス 106 に 4 バイトのデータとして出力する。

これらは、第7図 (A), (B) に示した通りである。デコード処理終了バイト数検出器 65 について説明する。

この部分は、本発明になる不定長命令 (S ビット付加したもの) を扱う場合の要部である。

デコード処理終了バイト数検出器 65 には、前述した如く、オペラント終了フラグ E を示す信号線 203、オペラント指定子のストップビット S を示す信号線 205 及びオペラント指定子のバイト数 (OS_n) を示す信号線 215 の 3 つの信号線が入力されており、信号線 214 へは、次のオペラント指定子のアドレスを示すため、DP 69 の加算値 DPINC_n をバイト単位で出力する。この場合のアルゴリズムは次の通りである。

E = 1 であれば、

DPINC_n = OS_n

これ以外で、S = 0 であれば、

DPINC_n = OS_n - 1

また、S = 1 であれば

DPINC_n = 0

すなわち、

(1) オペラントの終了フラグ E が "1" の場合は、該当命令の処理は終了したため、次の命令の先頭をさすように、DP レジスタ 69 がオペラント指定子のバイト数分 (OS_n) 加算されるよう信号線 214 へ出力される。

(2) (1)でなく、また終了フラグ S がセットされていない場合は、次のオペラント指定子が 1 バイトのダミーを先頭バイトにおいて、バス 104 へ出力させるため、(オペラント指定子のバイト数: OS_n) - 1 の値が加算されるよう信号線 214 へ出力される。

(3) (1)でなく、また終了フラグ S がセットされている場合は、DP レジスタ 69 が、そのまゝ

の値をとるよう、"0" が出力される。これによつて、同一のオペラント指定子を、次のオペラントに対する処理にも使用することになり、同一のオペラントが繰返し使用されることになる。

第8図は、デコード処理終了バイト数検出器 65 における上記アルゴリズムを実現するハードウェア構成を示している。

つまり、E = 1 の場合は、出力ゲート 301 が開き、バス 214 にはバス 215 の内容 OS_n が出力され、E = 0 の時は、S = 0 のとき出力ゲート 303 が開き、OS_n - 1 を出力し、また S = 1 のとき出力ゲート 302 が開き、"0" が出力されるようになつている。

本発明においては、E が "1" の時、S ビットが "1" でなければ、オペラント数より多いオペラント指定子が有ることを意味し、これをエラーとして検出しようとするものである。第8図ではアンドゲート 304 がオンとなり信号 105-3 を出力し、アドレス計算ユニット A U 28 に対し、

該当エラー発生を連絡するようになつてゐる。

アドレス計算ユニットAU28では、信号105-3を以下のユニット(オペランドフェッチユニットOFU29)に連絡し、最終的に実行ユニットEU30にエラー発生が連絡されるようになつてゐる。

尚第8図において、304, 305はインバータ、306, 307はアンドゲート、308は減算器である。

加算器71は、現在のDPレジスタ69の値と、信号線214の値を加算し、セレクタ70を介して、DPレジスタ69にセットすることで次のオペラント指定子のアドレスをバス102に出力することが可能となる。これにより、アライナALIG26は、次のオペラント指定子を第5図(A), (B)に示すフォーマットでバス104に出力できる。

一方、セレクタ70によつて、バス115の内容を選択してDPレジスタ69にセットすることで前述した分岐命令におけるDPレジスタ69の

繰返しデコードすることによりこれを実現しているが、これ以外にもデコードユニットから実行ユニットEU30に特別な信号を送り、先に求められたオペラントを繰返し利用するように指示してもよい。

図面の簡単な説明

第1図は本発明が適用されるデータ処理システムの基本的な概念図、第2図(A), (B)は本発明に使用される不定長命令のフォーマット及び、オペラント指定子のフォーマットを示す図、第3図は第1図の中央処理装置の具体的な実施例ブロック構成図、第4図は、本発明の要部である第3図のデコードユニットの具体的な実施例ブロック図、第5図は第4図の説明に用いられるオペラント指定子のフォーマット、第6図は第4図のオペラント情報ROM63の内容を説明するための説明図、第7図は第4図のオペラント指定子デコーダ66の動作説明に用いる説明図、第8図は、第4図のデコード処理終了バイト数検出器65のハード構成を示す実施例図である。

変更も可能となる。

尚加算器72は、DPレジスタ69の値に、該当オペラント指定子の長さを示す信号線215の値(OS₈)を加え、さらにキヤリー入力“1”を加えることで、デコードしているオペラント指定子の次のアドレスをバス116に出力する。

アドレス計算ユニットAU28は、バス116の内容をアドレス計算に使用するプログラムカウンタPCの値として利用する。

このように、本発明によれば、オペラント数とオペラント¹の個数の合理性チェックを行うことにより、エラー検出率を向上させることができる。

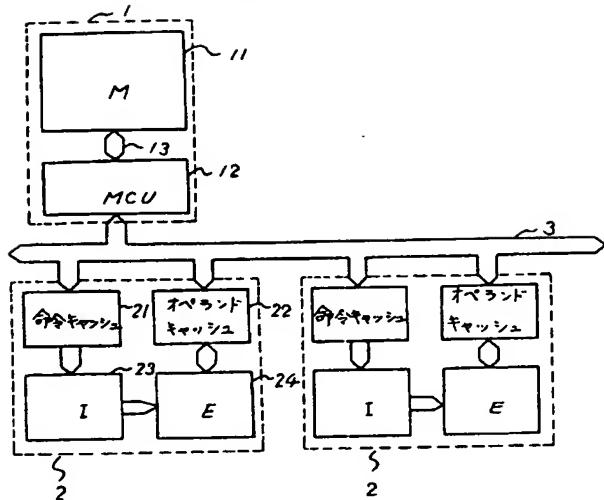
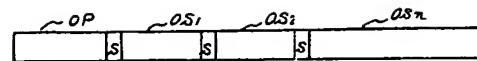
尚上記実施例においては、終了フラグSはオペラント指定子の最上位ビットに付加されているが、必ずしもこの部分に限定する必要はなく、オペラント指定子のどこかに終了フラグを設ければよい。

また、同一のオペラントを複数回繰返し、利用するのに、上記実現例では、DPレジスタ69の内容を更新させずに、同一のオペラント指定子を

27…デコードユニット、61…オペコードデコードユニット、63…オペラント情報ROM、65…デコード処理終了バイト数検出器、66…オペラント指定子デコーダ。

代理人弁理士 高橋明^{大高辨} 田中明子^{明士}

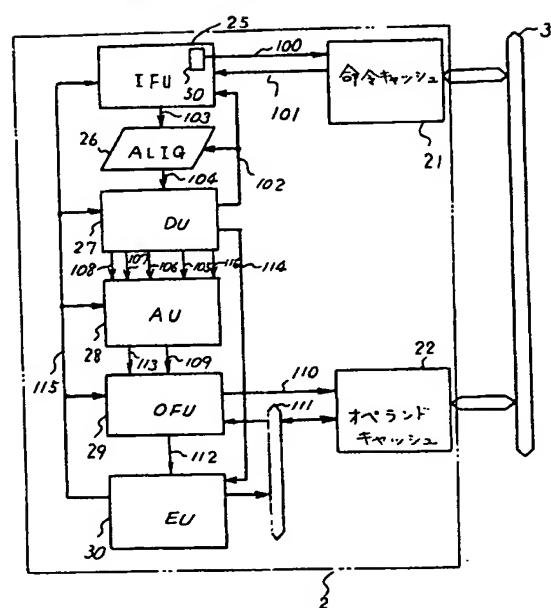
第1図

第2図
(A)第2図
(B)

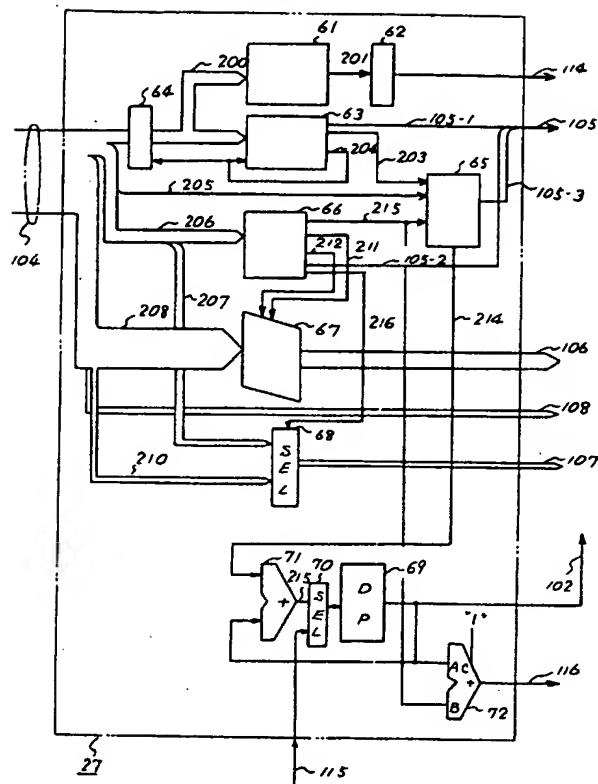
NO.	フォーマット	オペランド
1	S1000 Rn	Rn
2	S1001 Rn	(Rn)
3	S1010 Rn DISP8	(Rn+DISP8)
4	S1011 Rn DISP8	((Rn+DISP8))
5	S1100 Rn DISP16	(Rn+DISP16)
6	S1101 Rn DISP16	((Rn+DISP16))
7	S1110 0110 - Rn DISP8	(Rn+DISP8)
8	S1110 0111 - Rn DISP8	((Rn+DISP8))
9	S1110 0000 IM8	IM8
10	S1110 0001 IM16	IM16
11	S1110 0010 IM32	IM32
12	S1110 1010 DISP8	(PC+DISP8)
13	S1110 1011 DISP8	((PC+DISP8))
14	S1110 1100 DISP16	(PC+DISP16)
15	S1110 1101 DISP16	((PC+DISP16))
16	S1110 1110 DISP32	(PC+DISP32)
17	S1110 1111 DISP32	((PC+DISP32))
18	S1100 0000 Rx Rn	(Rn+RX-L)
19	S1100 0010 Rx Rn DISP8	(Rn+DISP8+RX-L)
20	S1100 0011 Rx Rn DISP8	((Rn+DISP8)+RX-L)
21	S1100 0100 Rx Rn DISP16	(Rn+DISP16+RX-L)
22	S1100 0101 Rx Rn DISP16	((Rn+DISP16)+RX-L)
23	S1100 0110 Rx Rn DISP32	(Rn+DISP32+RX-L)
24	S1100 0111 Rx Rn DISP32	((Rn+DISP32)+RX-L)

1バイト 1バイト 1バイト 1バイト 1バイト

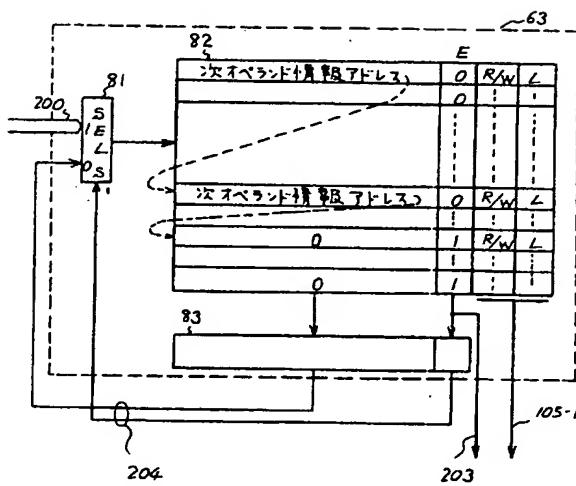
第3図



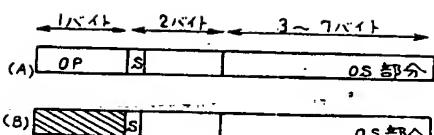
第 4 図



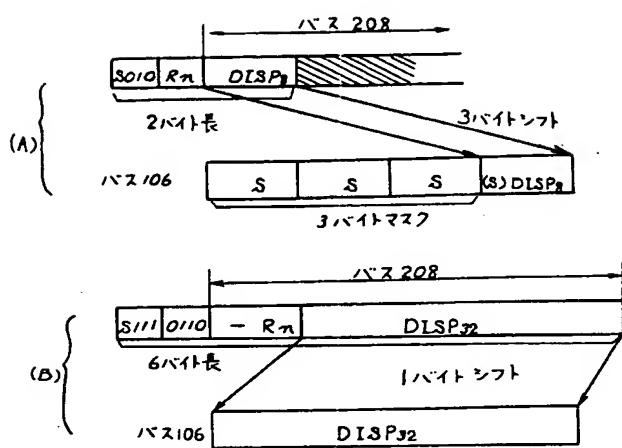
第 6 図



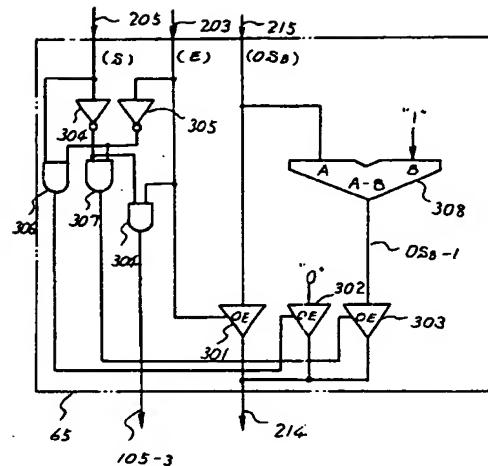
第 5 図



第 7 図



第 8 図



第 1 頁の続き

⑦発明者 井手寿之

日立市大みか町 5 丁目 2 番 1 号
株式会社日立製作所大みか工場
内

⑦発明者 加藤猛

日立市大みか町 5 丁目 2 番 1 号
株式会社日立製作所大みか工場
内

⑦発明者 中西宏明

日立市大みか町 5 丁目 2 番 1 号
株式会社日立製作所大みか工場
内

⑦発明者 河上哲也

日立市幸町 3 丁目 2 番 1 号 日立
エンジニアリング株式会社 内

⑦出願人 日立エンジニアリング株式会社
日立市幸町 3 丁目 2 番 1 号DOCKET NO: P2001,0304SERIAL NO: 10/694,591APPLICANT: Hartlieb et al.

LERNER AND GREENBERG P.A.

P.O. BOX 2480

HOLLYWOOD, FLORIDA 33022

TEL. (354) 325-1100